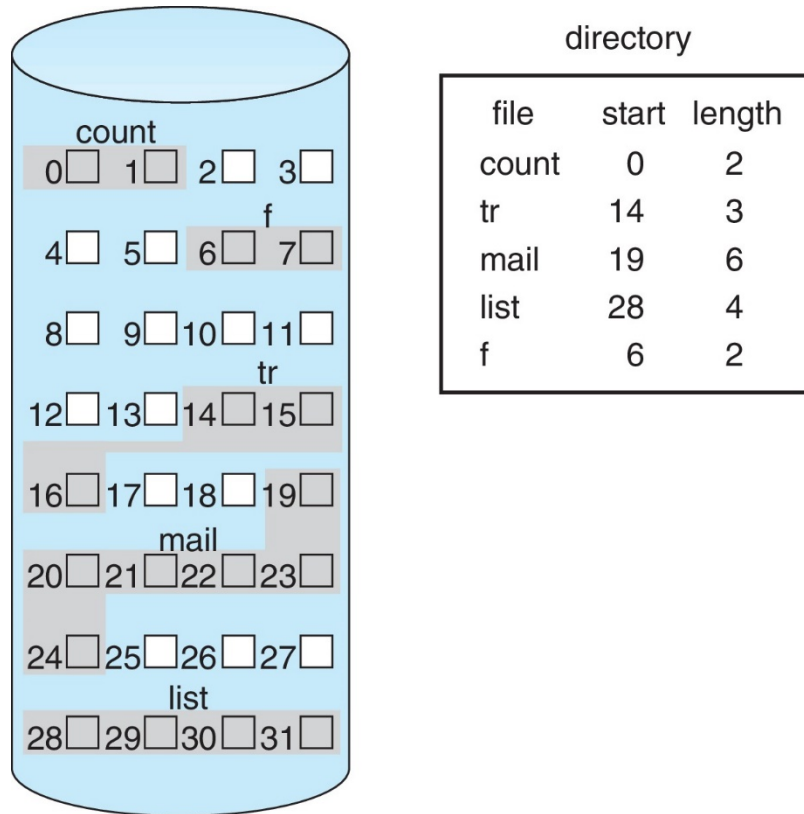


## Allocation Methods



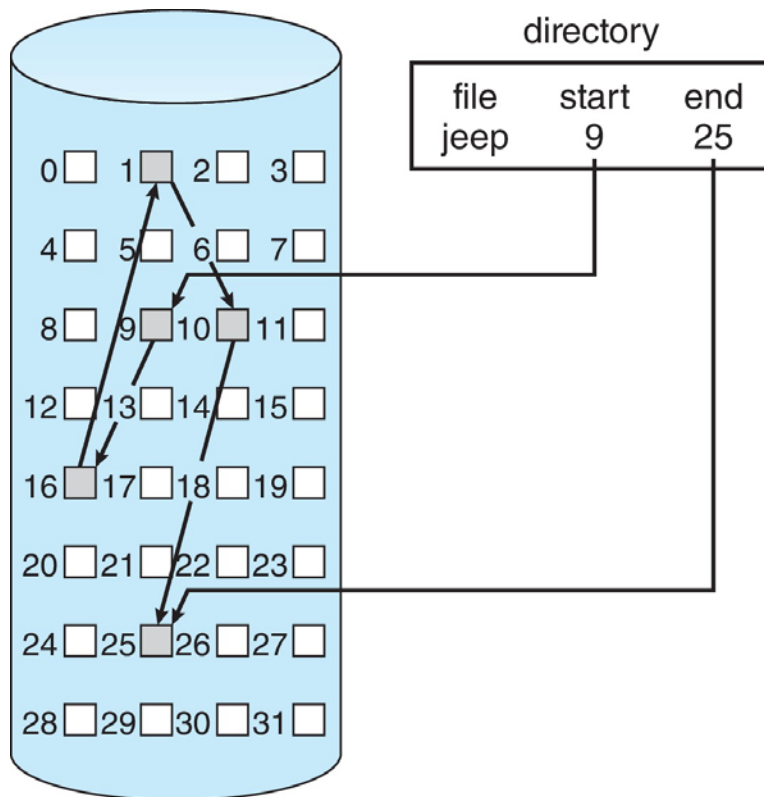
**Contiguous allocation** – each file occupies set of contiguous blocks

Best performance in most cases.

Simple – only starting location (block #) and length (number of blocks) are required.

Problems include finding space for file, knowing file size, need for compaction off-line (downtime) or on-line.

External and internal fragmentation.



**Linked allocation** – each file a linked list of blocks

Each block contains pointer to next block.

No external fragmentation.

Needs compaction.

No direct/random access

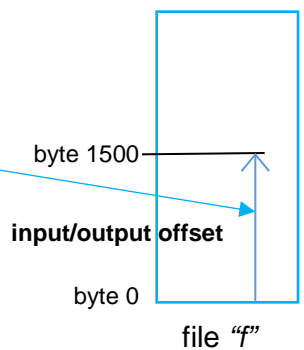
```
int fd=open (f, RD_ONLY);
```

```
...
```

```
lseek(fd, 1500, 0);
```

```
read (fd, buffer, 500);
```

```
...
```



directory entry

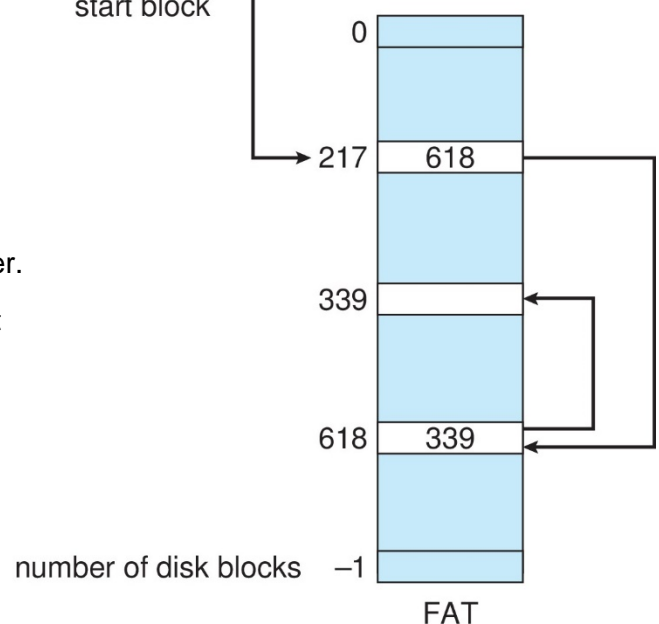
test	...	217
name		start block

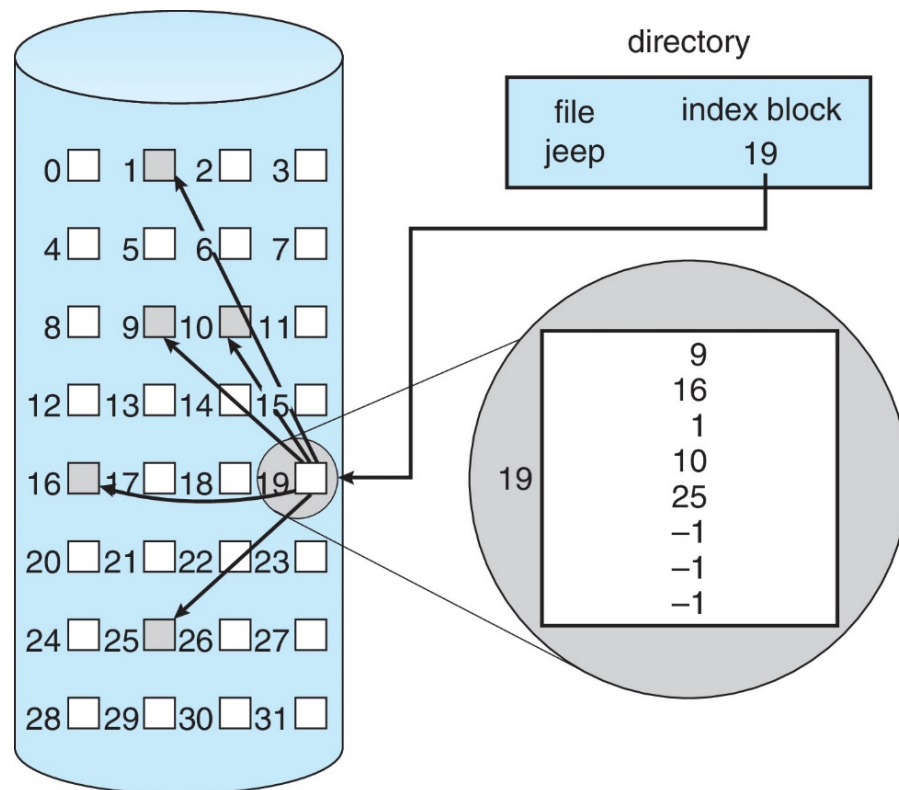
### FAT (File Allocation Table) variation

Beginning of volume has table, indexed by block number.

The FAT is cacheable (maintained in main memory -> it allows direct access).

New block allocation simple.





**Indexed Allocation** - Each file has its own index block(s) of pointers to its data blocks

Need index table for each file.

Direct access.

Dynamic access without external fragmentation, but have overhead of index block.